Discussion on "Bayesian Model Selection and Parameter Estimation in Extragalactic Astronomy" by Martin Weinberg

Phil Gregory
Physics and Astronomy
Univ. of British Columbia

Introduction

Martin Weinberg reported on the UMass Bayesian Inference Engine (BIE) package for model selection and parameter estimation in extragalactic astronomy.

The BIE philosophy is that there is no single best MCMC algorithm and develop a variety of MCMC algorithms augmented by different tools like parallel tempering, simulated annealing and differential evolution depending on the complexity of the problem.

My approach has been to attempt to fuse together the advantages of all of the above tools together with a genetic crossover operation in a single MCMC algorithm to facilitate the detection of a global maximum in probability. I call it fusion MCMC (FMCMC).

In applications to real exoplanet data the FMCMC algorithm has proved highly effective. For more details on FMCMC (previously named hybrid MCMC) and examples of exoplanet applications see my papers at

http://www.physics.ubc.ca/~gregory/gregory.html

So what are some of the useful lessons?

Outline

Fusion MCMC

The latest version of FMCMC for nonlinear model fitting incorporates:

Parallel tempering
Simulated annealing
Genetic algorithm

Each of these methods was designed to facilitate the detection of a global minimum in χ^2 .

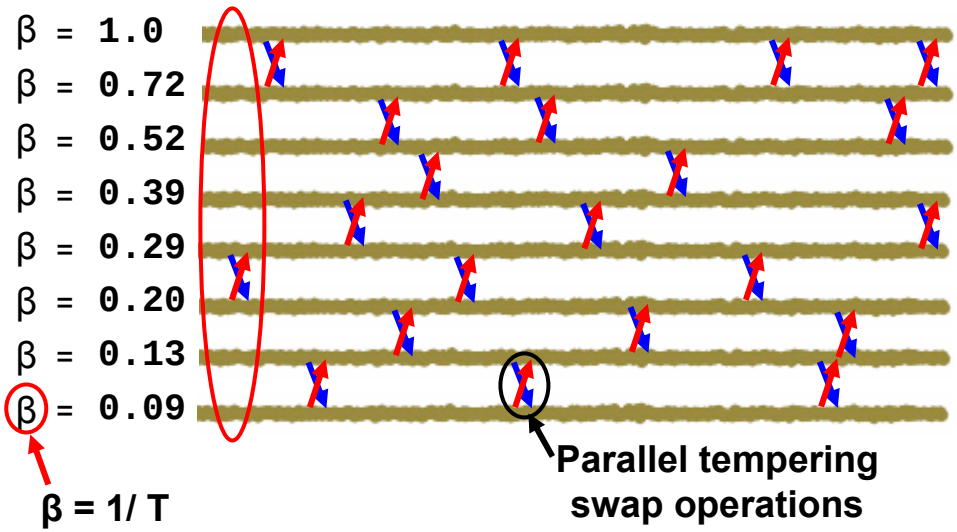
(in a Bayesian context: a global maximum of probability)

By combining all three we greatly increase the probability of realizing this goal.

This fusion has only been possible through the development of a unique adaptive control system. Among other things it automates the choice of an efficient set of MCMC proposal distributions even if the parameters are highly correlated.

Adaptive Fusion MCMC

8 parallel tempering Metropolis chains



Output at each iteration

parameters, logprior + β × loglike, logprior + loglike

parameters, logprior + β × loglike, logprior + loglike

parameters, logprior + β × loglike, logprior + loglike

parameters, logprior + β × loglike, logprior + loglike

parameters, logprior + β × loglike, logprior + loglike

parameters, logprior + β × loglike, logprior + loglike

parameters, logprior + β × loglike, logprior + loglike

parameters, logprior + β × loglike, logprior + loglike

The 8 parallel chains employ probability distributions of the kind

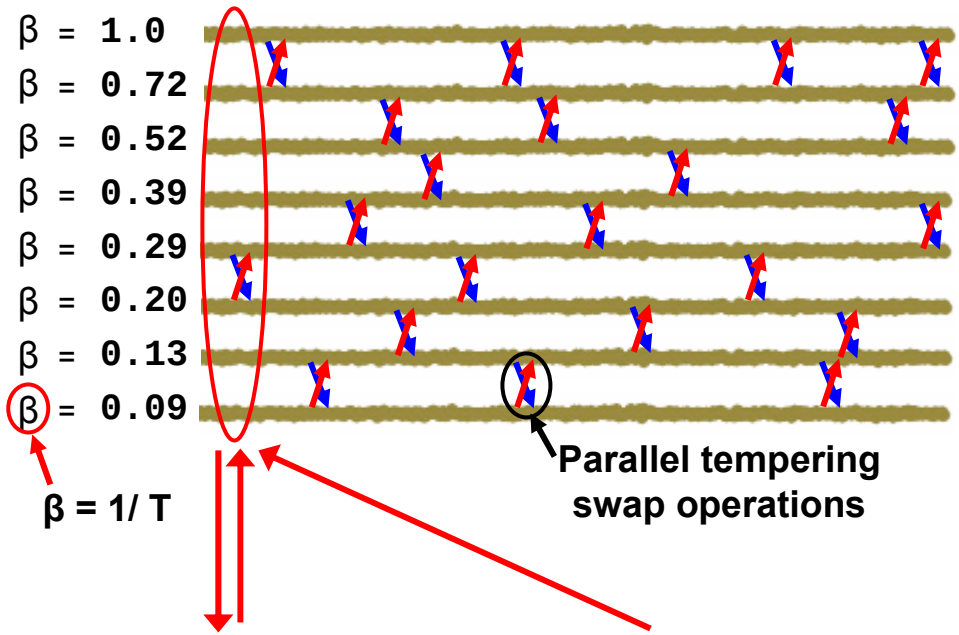
$$\pi(\mathbf{X}\,|\,\mathbf{D}, \mathbf{M}, \beta, \mathbf{I}) = \mathbf{p}(\mathbf{X}\,|\,\mathbf{M}, \mathbf{I})\ \mathbf{p}(\mathbf{D}\,|\,\mathbf{X}, \mathbf{M}, \mathbf{I})^{\beta} \quad (0 < \beta \leqslant 1)$$

β = 1 corresponds to our desired target distribution. The others correspond to progressively flatter probability distributions.

At intervals, a pair of adjacent chains are chosen at random and a proposal made to swap their parameter states. The swap allows for an exchange of information across the ladder of chains. In the low β chains, radically different configurations can arise, whereas at higher β, a configuration is given the chance to refine itself.

Adaptive Fusion MCMC

8 parallel tempering Metropolis chains



Output at each iteration

parameters, logprior + β × loglike, logprior + loglike

parameters, logprior + β × loglike, logprior + loglike

parameters, logprior + β × loglike, logprior + loglike

parameters, logprior + β × loglike, logprior + loglike

parameters, logprior + β × loglike, logprior + loglike

parameters, logprior + β × loglike, logprior + loglike

parameters, logprior + β × loglike, logprior + loglike

parameters, logprior + β × loglike, logprior + loglike

Anneal Gaussian proposal σ's → **Refine & update Gaussian proposal σ's**

2 stage proposal σ control system

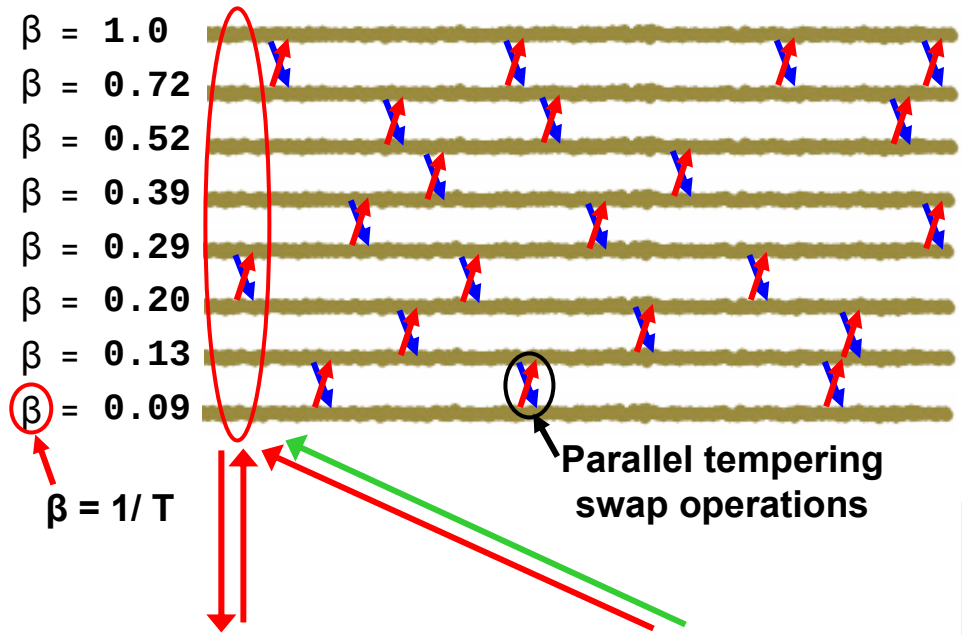
error signal =
(actual joint acceptance rate – 0.25)

Effectively defines burn-in interval

Portion of Control System that automates the selection of an efficient set of σ values for the independent Gaussian proposal distributions ('I' proposals).

Adaptive Fusion MCMC

8 parallel tempering Metropolis chains



Output at each iteration

parameters, logprior + β × loglike, logprior + loglike

parameters, logprior + β × loglike, logprior + loglike

parameters, logprior + β × loglike, logprior + loglike

parameters, logprior + β × loglike, logprior + loglike

parameters, logprior + β × loglike, logprior + loglike

parameters, logprior + β × loglike, logprior + loglike

parameters, logprior + β × loglike, logprior + loglike

parameters, logprior + β × loglike, logprior + loglike

Monitor for parameters with peak probability

Peak parameter set:
If (logprior + loglike) > previous best by a threshold then update and reset burn-in

Anneal Gaussian proposal σ's → Refine & update Gaussian proposal σ's

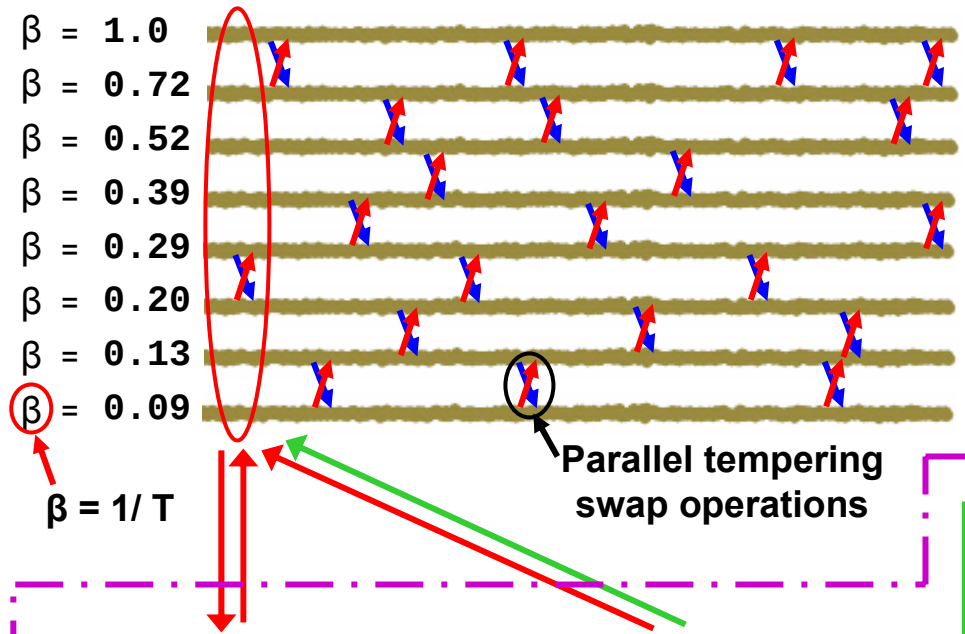
2 stage proposal σ control system

error signal =
(actual joint acceptance rate – 0.25)

Effectively defines burn-in interval

Adaptive Fusion MCMC

8 parallel tempering Metropolis chains



Output at each iteration

parameters, logprior + β × loglike, logprior + loglike

parameters, logprior + β × loglike, logprior + loglike

parameters, logprior + β × loglike, logprior + loglike

parameters, logprior + β × loglike, logprior + loglike

parameters, logprior + β × loglike, logprior + loglike

parameters, logprior + β × loglike, logprior + loglike

parameters, logprior + β × loglike, logprior + loglike

parameters, logprior + β × loglike, logprior + loglike

Monitor for parameters with peak probability

Peak parameter set:
If (logprior + loglike) > previous best by a threshold then update and reset burn-in

Anneal Gaussian proposal σ's

Refine & update Gaussian proposal σ's

2 stage proposal σ control system

error signal =
(actual joint acceptance rate – 0.25)

Effectively defines burn-in interval

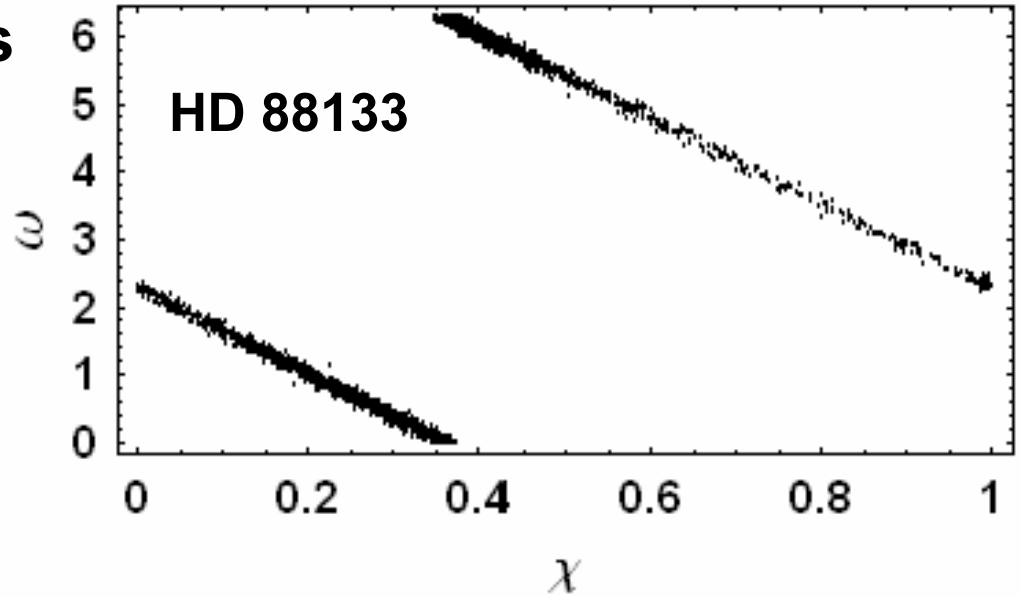
Genetic algorithm

Every 10th iteration perform gene crossover operation to breed larger (logprior + loglike) parameter set.

MCMC adaptive control system

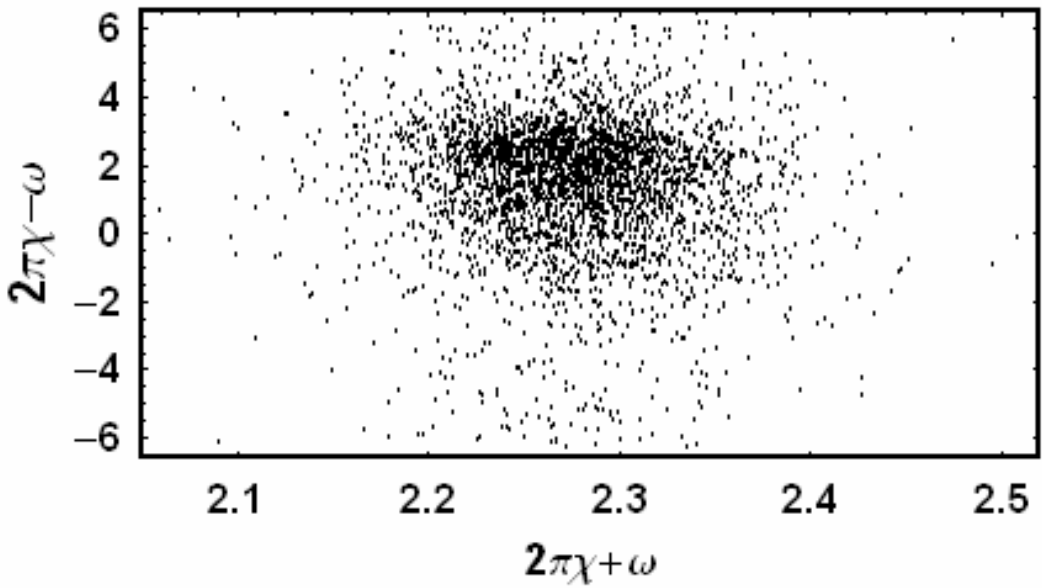
Highly correlated parameters

Exoplanet example: for low eccentricity orbits the parameters ω and χ are not separately well determined. This shows up as a strong correlation between ω and χ as shown on the right.



One option re-parameterization

The combination $2\pi\chi+\omega$ is well determined for all eccentricities. Although $2\pi\chi-\omega$ is not well determined for low eccentricities, it is at least orthogonal to $2\pi\chi+\omega$ as the lower figure demonstrates.



Another option

Algorithm learns about the parameter correlations during the burn-in and generates proposals with these statistical correlations.

How to deal with highly correlated parameters

One solution used by Weiner and others is Differential Evolution Markov Chain (DE-MC) (TerBraak 2006). DE-MC is a population MCMC algorithm in which multiple chains are run in parallel, typically from 15 to 40.

The proposed jumps are simply a fixed multiple of the differences of two random parameter vectors that are currently in the population.

DE-MC solves an important problem in MCMC, namely that of choosing an appropriate scale and orientation for the jumping distribution.

My fusion MCMC algorithm already runs parallel tempering chains to avoid becoming trapped in a local probability maximum. To increase the number of chains by a further factor of 15 to 40, to accomplish DE-MC, would not be practical.

Necessity being the mother of invention, I developed a new approach that automatically achieves efficient MCMC sampling in highly correlated parameter spaces without the need for additional chains.

How to deal with highly correlated parameters

Using only independent Gaussian proposals the ('I' scheme) the σ's need to be very small for any proposal to be accepted and consequently convergence is very slow.

Learn about parameter correlations during burn-in

The accepted 'I' proposals will generally cluster along the correlation path so every 2nd accepted 'I' proposal is appended to a correlated sample buffer (separate buffer for each tempering level).

Only the 300 most recent additions to the buffer are retained.

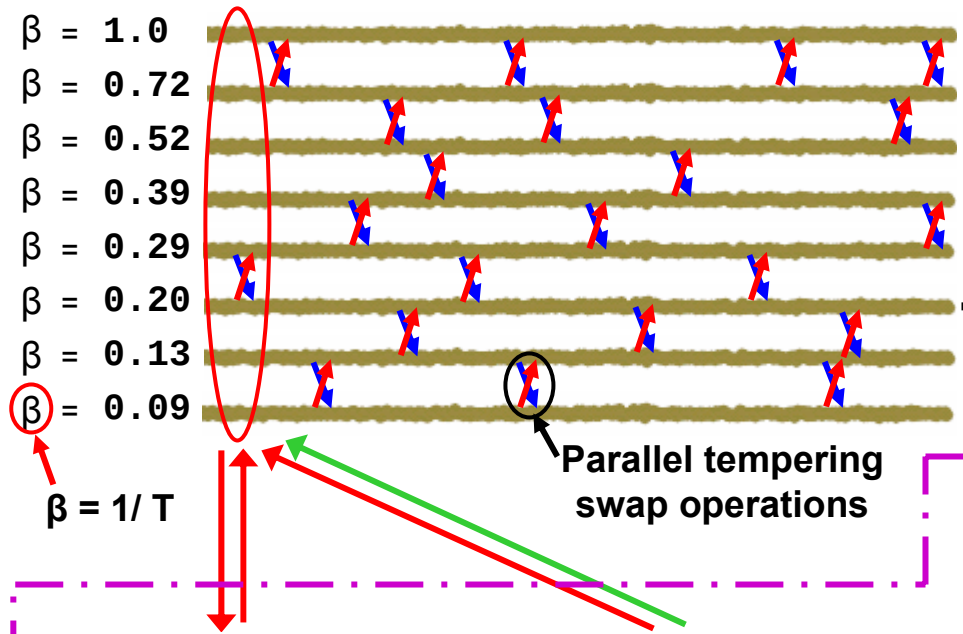
A 'C' proposal is generated using the difference between a pair of randomly selected samples drawn from the correlated sample buffer (for that tempering level), after multiplication by a constant.

Value of constant is computed automatically by another module which ensures that the 'C' proposal acceptance rate is close to 25%.

Adaptive Fusion MCMC

Automatic proposal scheme that learns about parameter correlations during burn-in (for each chain)

8 parallel tempering Metropolis chains



new parameter value

from proposal C

from proposal I

repeat

add every second to buffer

latest 300 values

difference of random pairs

multiply by constant <1

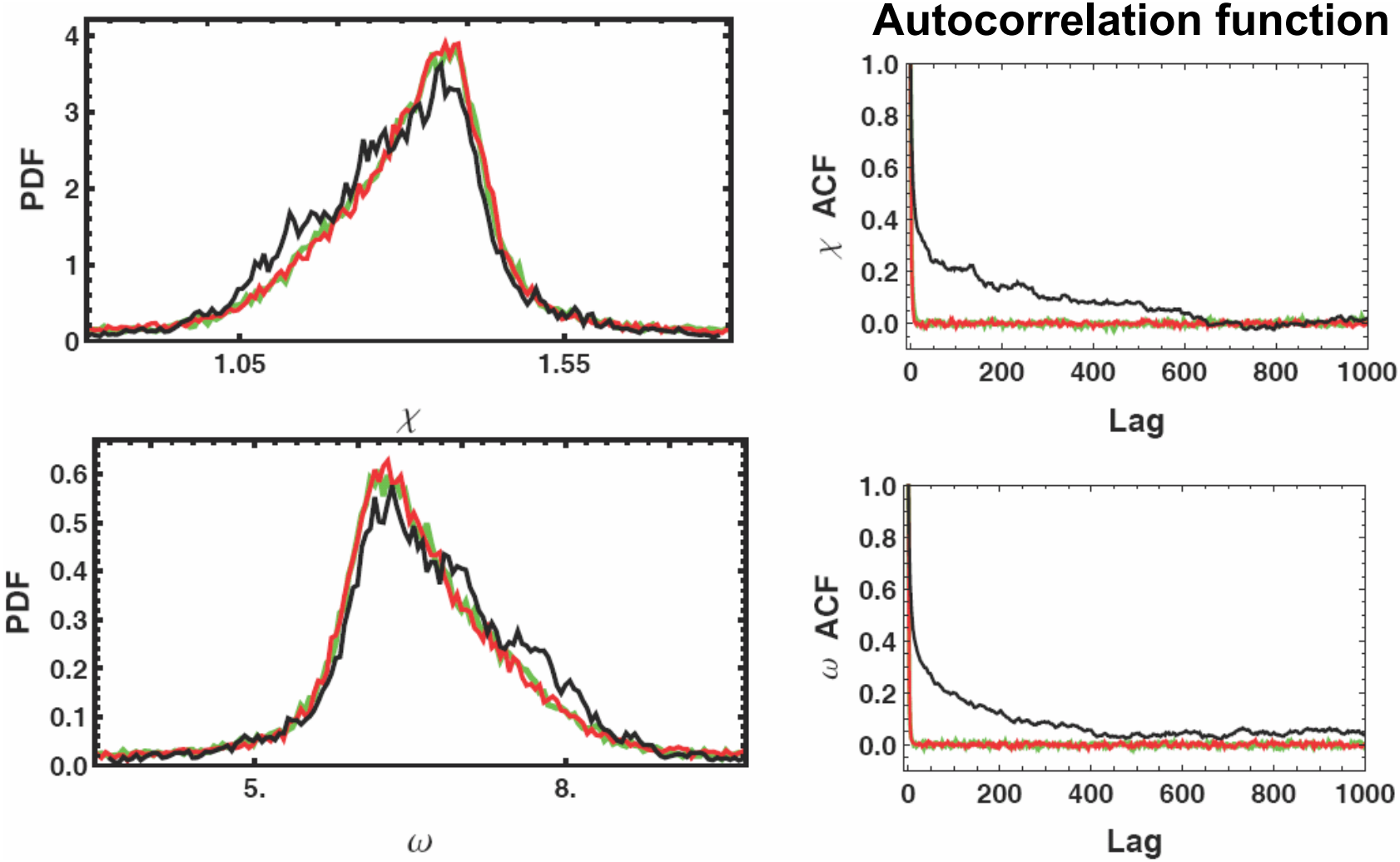
Proposal I
Independent Gaussian proposal scheme employed 50% of the time

Proposal C
Proposal distribution with built in parameter correlations used 50% of the time

During burn-in control system adjusts constant so acceptance rate from Proposal C = 25 %

MCMC adaptive control system

Testing ‘C’ proposal scheme



Left panels show the MCMC marginal probability distributions for parameters χ and ω. Right panels show their MCMC autocorrelation functions.

Black trace = search in χ and ω using only ‘I’ proposals.

Red trace = search using both ‘I’ and ‘C’ proposals.

Green trace = ‘I’ search using transformed orthogonal coordinates.

Summary of Automatic ‘C’ proposal features

With very little computational overhead, the ‘C’ proposals provide the scale and direction for efficient jumps in a correlated parameter space with no additional chains.

The final proposal distribution is a random selection of ‘I’ and ‘C’ proposals such that each is employed 50% of the time.

The combination ensures that the whole parameter space can be reached and that the FMCMC chain is aperiodic.

The parallel tempering feature operates as before to avoid becoming trapped in a local probability maximum.

Noise model

Weinberg concludes that data-model comparison without an accurate error model is likely to be erroneous.

I find it very useful to incorporate extra noise parameter, s, of unknown magnitude, added in quadrature to the known measurement uncertainties σ_i .

$$\text{variance}_i = \sigma_i^2 + s^2$$

Marginalizing *s* has the desirable effect of treating anything in the data that can't be explained by the model and known measurement errors as noise, leading to more conservative estimates of the parameters.

If there is no extra noise then the posterior probability distribution for s will peak at s = 0.

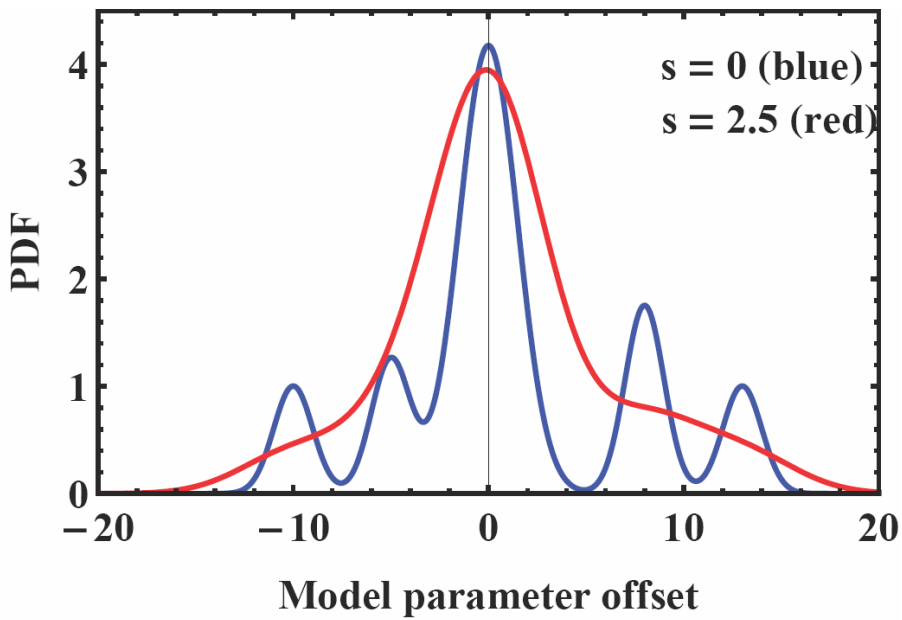
Annealing due to extra noise term, s

Inclusion of an extra noise term of unknown magnitude also gives rise to an annealing operation when the Markov chain is started far from the best-fit values.

If only known observational errors are included, the posterior probability distribution is often very "rough" with many local maxima throughout parameter space.

When *s* is included, Bayesian Markov chain automatically inflates *s* to include anything in the data that cannot be accounted for by the model with the current set of parameters and the known measurement errors.

This results in a smoothing out of the posterior surface and allows the Markov chain to explore the parameter space more quickly. The chain begins to decrease the value of *s* as it settles in near the best-fit parameters. This behavior is similar to simulated annealing, but does not require choosing a cooling scheme.



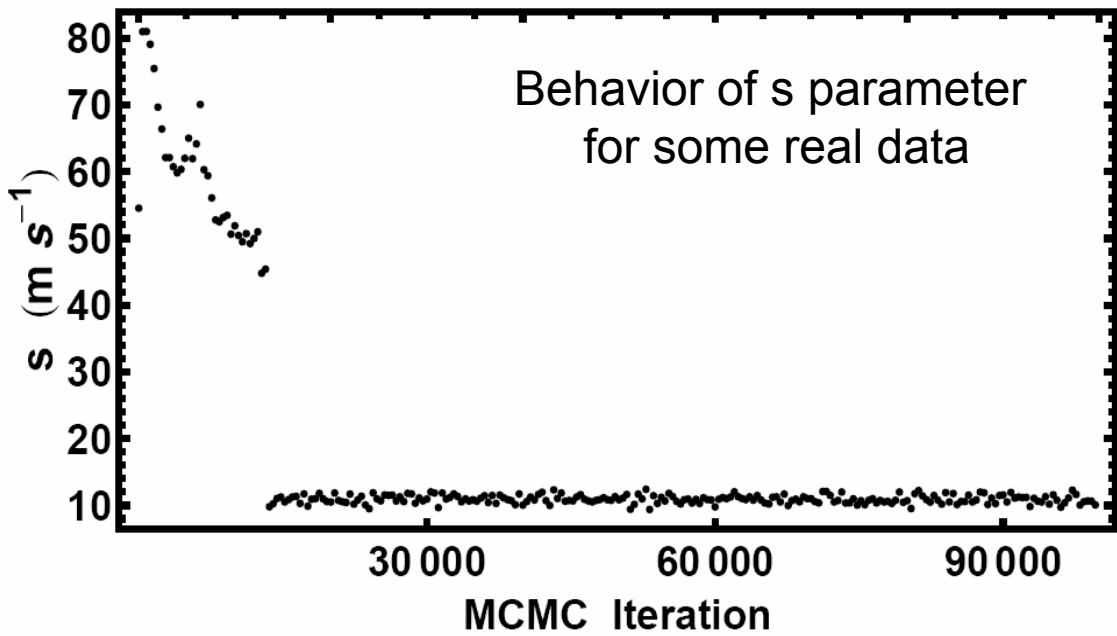
Annealing due to extra noise term, s

Inclusion of an extra noise term of unknown magnitude also gives rise to an annealing operation when the Markov chain is started far from the best-fit values.

If only known observational errors are included, the posterior probability distribution is often very "rough" with many local maxima throughout parameter space.

When *s* is included, Bayesian Markov chain automatically inflates *s* to include anything in the data that cannot be accounted for by the model with the current set of parameters and the known measurement errors.

This results in a smoothing out of the posterior surface and allows the Markov chain to explore the parameter space more quickly. The chain begins to decrease the value of *s* as it settles in near the best-fit parameters. This behavior is similar to simulated annealing, but does not require choosing a cooling scheme.



Model Selection

One of the great strengths of Bayesian analysis is the built-in Occam's razor. More complicated models contain larger numbers of parameters and thus incur a larger Occam penalty, which is automatically incorporated in a Bayesian model selection analysis in a quantitative fashion.

MCMC achieves integration to within a proportionality constant which is fine for parameter estimation. For model selection we need to know the proportionality constant so must employ other integration techniques.

Bayesian model selection is easy in concept but becomes progressively more difficult to compute as the number of model parameters increases. It is thus important to compare results from different methods.

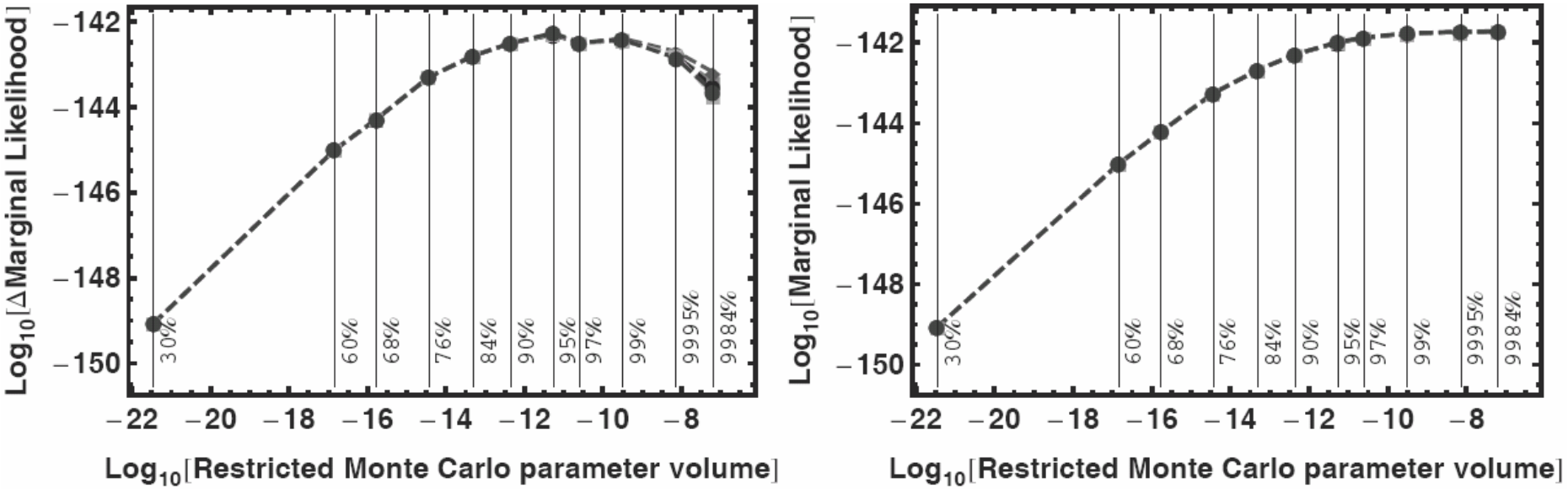
Model Selection

One method I employ to estimate the marginal likelihoods is Nested Restricted Monte Carlo (NRMC) integration (Gregory & Fischer 2010). For large parameter spaces, Monte Carlo (MC) integration is hopelessly inefficient in exploring the whole prior parameter range. The fraction of the prior volume containing significant probability rapidly declines as the number of dimensions increase.

In Restricted MC (RMC) this is less of a problem because the volume of parameter space sampled is greatly restricted to a region delineated by the outer borders of the marginal distributions of the parameters.

In Nested RMC (NRMC) integration, multiple boundaries are constructed based on credible regions ranging from 30% to > 99%, as needed. We are able to compute the contribution to the total integral from each nested interval and sum these contributions. For example, for the interval between the 30% and 60% credible regions, we generate random parameter samples within the 60% region and reject any sample that falls within the 30% region. Using the remaining samples we can compute the contribution to the NRMC integral from that interval.

Nested Restricted Monte Carlo Integration



The left panel shows the contributions from the individual intervals for 5 repeats of the NRMC evaluation for the 3 planet model. The right panel shows the summation of the individual contributions versus the volume of the credible region.

The 9995% boundary is defined as follows. Let XU99 and XL99 correspond to the upper and lower boundaries of the 99% credible region, for a particular parameter. Similarly, XU95 and XL95 are the upper and lower boundaries of the 95% credible region for the parameter.

Then XU9995 = XU99 +(XU99−XU95),

XL9995 = XL99 +(XL99−XL95).

Similarly, XU9984 = XU99 + (XU99 −XU84).

Comparison of NRMC & Ratio Estimator Marginal Likelihoods

For three planet models (17 unknown parameters)

Star	NRMC Estimator / Ratio Estimator (improved version)
HD11964*	0.90
(* 1 of the signals is a suspected artifact)	
47 UMa	0.75
Gliese 581	1.01

For four planet models (22 unknown parameters)

Gliese 581	0.016

For large numbers of parameters the NRMC method is expected to underestimate the marginal likelihood and the RE method to overestimate (potential to pay too much attention to the mode as each integrand involves the square of the posterior density).

This appears to be happening by the time we reach 22 parameters.

Ratio Estimator Marginal Likelihood (Jim Berger suggestion)

Start by re-writing Bayes theorem

$$p\,(D \mid M)\; p\left(\vec{\theta} \mid D,\, M\right) \;=\; p\left(\vec{\theta} \mid M\right)\; p\left(D \mid \vec{\theta},\, M\right)$$

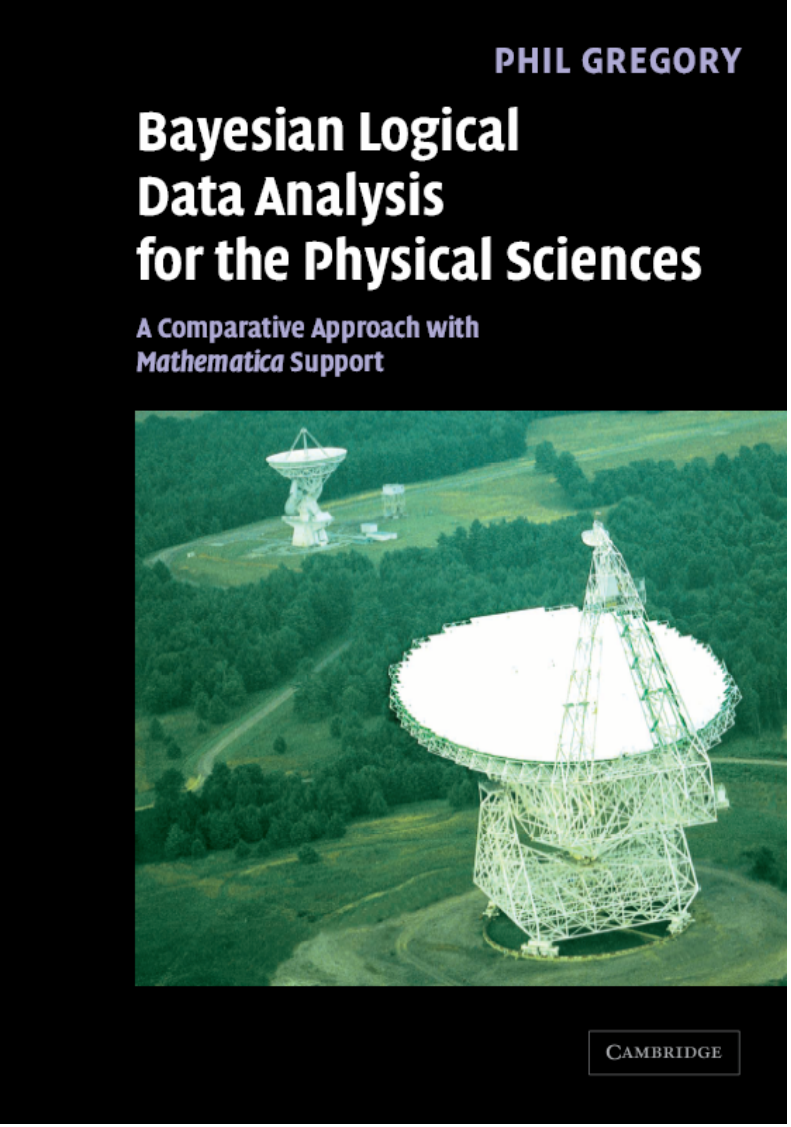
Multiply by function $h\,(\vec{\theta})$ = a mixture of multivariate Normals that approximate the MCMC samples of $p\left(\vec{\theta} \mid D,\, M\right)$, and integrate over $\vec{\theta}$.

$$p\,(D \mid M) \int p\left(\vec{\theta} \mid D,\, M\right) h\,(\vec{\theta})\, d\vec{\theta} = \int p\left(\vec{\theta} \mid M\right)\; p\left(D \mid \vec{\theta},\, M\right) h\,(\vec{\theta})\, d\vec{\theta}$$

Then the marginal likelihood ratio estimator (RE) is given by

$$p\,(D \mid M)_{RE} \;\approx\; \frac{\frac{1}{N_i} \sum_{i=1}^{N_i} p\left(\vec{\theta}_i \mid M\right)\; p\left(D \mid \vec{\theta}_i,\, M\right)}{\frac{1}{N_j} \sum_{j=1}^{N_j} h\,(\vec{\theta}_j)}$$

The $\vec{\theta}_i$ samples are random drawns from $h\,(\vec{\theta})$ and the $\vec{\theta}_j$ samples are random drawns from the MCMC samples of $p\left(\vec{\theta} \mid D,\, M\right)$.



Chapters

1. Role of probability theory in science
2. Probability theory as extended logic
3. The how-to of Bayesian inference
4. Assigning probabilities
5. Frequentist statistical inference
6. What is a statistic?
7. Frequentist hypothesis testing
8. Maximum entropy probabilities
9. Bayesian inference (Gaussian errors)
10. Linear model fitting (Gaussian errors)
11. Nonlinear model fitting
12. Markov chain Monte Carlo
13. Bayesian spectral analysis
14. Bayesian inference (Poisson sampling)

Resources and solutions
This title has free Mathematica based support software available

Introduces statistical inference in the larger context of scientific methods, and includes 55 worked examples and many problem sets.